Distributed Algorithms Coursework 1 - Report

Local, Docker and Distributed Machine Setup

Local

Local tests were run on Mac laptops with four cores.

Docker

Distributed Machine

Lab machines

System 1

Locally

Test No.	Max messages	Timeout	Output
1	1000	3000	2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	2: {90021, 281994} {90021, 288464} {90020, 90020} {90020, 283556} {90020, 88873} 1: {288465, 281994} {288464, 288464} {288464, 90021} {288464, 283556} {288464, 88873} 3: {283556, 281993} {283556, 288464} {283556, 90020} {283556, 283556} {283556, 88872} 0: {281994, 281994} {281994, 288464} {281994, 90021} {281993, 283556} {281993, 88873} 4: {88873, 281993} {88873, 288464} {88873, 90020} {88872, 283556} {88872, 88872}

Docker

Test No.	Max messages	Timeout	Output
1	1000	3000	2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	2: {32123, 6887} {32123, 8196} {32123, 32123} {32123, 4334} {32123, 3535} 4: {30026, 6886} {30025, 8857} {30025, 24054} {30025, 24611} {30025, 30025} 3: {24612, 6886} {24611, 8857} {24611, 29747} {24611, 24611} {24611, 29075} 1: {8858, 6887} {8858, 8858} {8858, 31994} {8857, 24611} {8857, 28896} 0: {6887, 6887} {6887, 8858} {6887, 32087} {6886, 24612} {6886, 29866}

System 2

Locally

Test No.	Max messages	Timeout	Output
1	1000	3000	3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	4: {72423, 27492} {72422, 47197} {72422, 34670} {72422, 33553} {72422, 30136} 0: {49277, 34640} {49277, 64599} {49276, 41701} {49276, 42357} {49276, 37136} 3: {58272, 37008} {58272, 69912} {58272, 44071} {58272, 45369} {58272, 39505} 1: {141405, 17445} {141405, 25714} {141405, 23648} {141405, 23465} {141405, 20446} 2: {72308, 30606} {72308, 54556} {72307, 37588} {72307, 37523} {72307, 33246}

In the first test case, all messages are broadcast and received. The sending and receiving of messages must have stopped before the timeout.

However in the second test case the maximum number of messages is not sent, as the sending process times out before all messages can be sent. In all cases more messages are sent from one peer to another than are received. For example, peer1 sends 141405 messages to peer0 but only receives 64599. This is not due to lossy sending, but because of our interleaving strategy:

- Interleaving strategy: Check if a message has been received. If not, send another message to the next peer (cyclically) and repeat.

This means that on some loops no message is received. Perhaps if we included a wait period for the next message before sending another we would see more equal values for sent and received.

Finally, far fewer messages are received when PL links are used.

Docker

Test No.	Max messages	Timeout	Output
1	1000	3000	3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	1: {54818, 7213} {54818, 13269} {54818, 6671} {54817, 3445} {54817, 7634} 4: {33137, 10474} {33137, 17275} {33137, 10165} {33136, 7459} {33136, 16224} 3: {47392, 2431} {47391, 3078} {47391, 2307} {47391, 1573} {47391, 5616} 0: {42041, 3971} {42041, 5766} {42041, 3115} {42040, 1937} {42040, 5626} 2: {69932, 7864} {69932, 13862} {69932, 8479} {69932, 4438} {69932, 11776}

On average less are sent and received when on Docker, likely due to the overhead.

System 3

Locally

Test No.	Max messages	Timeout	Output
1	1000	3000	1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	0: {220793, 1822} {220793, 1147} {220793, 481} {220793, 810} {220793, 1425} 4: {184181, 1822} {184181, 1159} {184181, 481} {184181, 824} {184181, 1425} 1: {255701, 669} {255701, 50} {255701, 34} {255701, 33} {255701, 375} 2: {194664, 1409} {194664, 314} {194664, 143} {194664, 145} {194664, 1011} 3: {229854, 4523} {229854, 6286} {229854, 3361} {229854, 6185} {229854, 4144}

Again, test 1 indicates that when enough time is provided (relative to max messages) all messages are sent and received.

In contrast to System2, far more messages are sent than are received. This is because Best Effort Broadcast is a different interleaving to System2:

- Best Effort Broadcast: Check if a message has been received. If not, send a message to every peer and repeat.

Hence the ratio of time spent sending messages to other peers and time spend receiving them has increased, explaining this difference.

Docker

Test No.	Max messages	Timeout	Output
1	1000	3000	2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
2	10000000	3000	3: {404216, 0} {404216, 0} {404216, 0} {404216, 0} {404216, 0} 0: {446376, 0} {446376, 0} {446376, 0} {446376, 0} {446376, 0} 4: {463456, 0} {463456, 0} {463456, 0} {463456, 0} {463456, 0} 1: {399327, 0} {399327, 0} {399327, 0} {399327, 0} {399327, 0} 2: {423217, 0} {423217, 0} {423217, 0} {423217, 0} {423217, 0}

System 4

Locally

Firstly, with only 1000 messages maximum:

Test No.	Max messages	Timeout	Reliability	Output
1	1000	3000	0	1: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 2: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 0: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 4: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 3: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0}
2	1000	3000	100	1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
3	1000	3000	50	1: {1000, 518} {1000, 505} {1000, 489} {1000, 493} {1000, 521} 3: {1000, 505} {1000, 497} {1000, 501} {1000, 496} {1000, 508} 0: {1000, 513} {1000, 494} {1000, 475} {1000, 479} {1000, 512} 2: {1000, 531} {1000, 509} {1000, 482} {1000, 497} {1000, 499} 4: {1000, 474} {1000, 513} {1000, 476} {1000, 497} {1000, 523}

Now with 10000000 messages maximum:

Test No.	Max messages	Timeout	Reliability	Output
1	10000000	3000	0	1: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 2: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 0: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 4: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 3: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0}
2	10000000	3000	100	1: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 2: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 4: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 0: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} 3: {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000} {1000, 1000}
3	10000000	3000	50	4: {242940, 216} {242940, 78} {242940, 72} {242940, 1068} {242940, 1403} 2: {224078, 1760} {224078, 1272} {224078, 1477} {224078, 4673} {224078, 2723} 1: {145257, 2131} {145257, 3358} {145257, 1829} {145257, 5122} {145257, 3232} 3: {190555, 2536} {190555, 4965} {190555, 2302} {190555, 5692} {190555, 3578} 0: {164838, 623} {164838, 92} {164838, 374} {164838, 2248} {164838, 1900}

Clearly, a reliability of 0 always results in no messages being. Additionally, a reliability of 100 gives identical results to that of System 3. Hence, tests of reliability 0 and 100 will not be run with different setups and systems.
Comparing reliabilities of 50, we see that the ratio of messages sent to received is much higher (around double, as expected) in the case where max messages is 1000, than where max messages is 10000000. This is because messages are broadcast n times faster (where n is the number of peers) than are received, and with only 1000 messages there is enough time to process the large backlog of messages after broadcasting is complete.

Docker

Test No.	Max messages	Timeout	Reliability	Output
1	1000	3000	50	2: {1000, 507} {1000, 490} {1000, 502} {1000, 509} {1000, 501} 0: {1000, 477} {1000, 511} {1000, 498} {1000, 478} {1000, 508} 1: {1000, 509} {1000, 499} {1000, 490} {1000, 517} {1000, 493} 3: {1000, 161} {1000, 477} {1000, 477} {1000, 507} {1000, 503} 4: {1000, 505} {1000, 495} {1000, 528} {1000, 510} {1000, 508}
2	10000000	3000	50	2: {378638, 0} {378638, 0} {378638, 0} {378638, 0} {378638, 0} 1: {353665, 0} {353665, 0} {353665, 0} {353665, 0} {353665, 0} 3: {407289, 0} {407289, 0} {407289, 0} {407289, 0} {407289, 0} 0: {300253, 0} {300253, 0} {300253, 0} {300253, 0} {300253, 0} 4: {328679, 0} {328679, 0} {328679, 0} {328679, 0} {328679, 0}

System4 behaves differently on Docker than locally for test 2 (though test 1 behaves similarly). Here, no messages are received at all.

System 5

Locally

Test No.	Max messages	Timeout	Reliability	Output
1	1000	3000	50	3: {0, 0} {0, 0} {0, 0} {0, 0} {0, 0} 4: {1000, 523} {1000, 516} {1000, 461} {1000, 0} {1000, 526} 1: {1000, 497} {1000, 521} {1000, 507} {1000, 0} {1000, 505} 0: {1000, 478} {1000, 513} {1000, 500} {1000, 0} {1000, 506} 2: {1000, 516} {1000, 497} {1000, 479} {1000, 0} {1000, 524}

2	10000000	3000	50	3: {393, 0} {393, 0} {393, 0} {393, 0} {393, 0} 1: {214726, 2959} {214726, 4623} {214726, 2178} {214726, 0} {214726, 3166} 4: {276731, 102} {276731, 41} {276731, 131} {276731, 0} {276731, 627} 0: {303203, 2959} {303203, 4550} {303203, 2177} {303203, 0} {303203, 3105} 2: {493496, 0} {493496, 0} {493496, 0} {493496, 0} {493496, 26}

As expected, Peer3 sends and receives no messages in test 1. The 5 millisecond kill time is short enough that it has no chance to send or receive messages. The other peers behave as in System4. Surprisingly, in test 2 Peer3 manages to send 393 messages to every other peer before terminating. Furthermore, none of these messages are received by other peers.

System 6

Locally

Test No.	Max messages	Timeout	Reliability	Output
1	1000	3000	100	3: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 1: {1000, 972} {1000, 974} {1000, 980} {1000, 0} {1000, 998} 4: {1000, 972} {1000, 974} {1000, 980} {1000, 0} {1000, 998} 2: {1000, 972} {1000, 974} {1000, 980} {1000, 0} {1000, 998} 0: {1000, 972} {1000, 974} {1000, 980} {1000, 0} {1000, 998}
2	1000	3000	50	3: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 0: {1000, 501} {1000, 867} {1000, 914} {1000, 0} {1000, 1125} 2: {1000, 971} {1000, 837} {1000, 486} {1000, 0} {1000, 1109} 1: {1000, 858} {1000, 507} {1000, 935} {1000, 0} {1000, 1121} 4: {1000, 979} {1000, 852} {1000, 1059} {1000, 0} {1000, 512}

Evidently, our solution is buggy since some peers receive more messages than the maximum possible number. We were expecting that most messages would reach all peers despite low reliability of PL links, since the network is fully connected and messages have multiple chances to reach peers with Eager broadcast.

Docker

Test No.	Max messages	Timeout	Reliability	Output
1	1000	3000	100	3: {1000, 0} {1000, 0} {1000, 0} {1000, 0} {1000, 0} 1: {1000, 995} {1000, 995} {1000, 992} {1000, 0} {1000, 956} 0: {1000, 995} {1000, 995} {1000, 992} {1000, 0} {1000, 956} 4: {1000, 995} {1000, 995} {1000, 992} {1000, 0} {1000, 956} 2: {1000, 995} {1000, 995} {1000, 992} {1000, 0} {1000, 956}
2	1000	3000	50	3: {461, 0} {461, 0} {461, 0} {461, 0} {461, 0} 0: {1000, 493} {1000, 927} {1000, 1062} {1000, 0} {1000, 845} 2: {1000, 839} {1000, 1085} {1000, 501} {1000, 0} {1000, 913} 1: {1000, 824} {1000, 503} {1000, 1044} {1000, 0} {1000, 970} 4: {1000, 759} {1000, 971} {1000, 1105} {1000, 0} {1000, 500}